

# ANALISI API REST PROGETTO LABORATORIO

## TABELLE

```
CREATE TABLE utenti(  
  idUtente int not null PRIMARY KEY auto_increment,  
  username varchar(30)not null unique,  
  nome varchar(30)not null,  
  cognome varchar(30)not null,  
  password varchar(32)not null,  
  codiceFiscale varchar(16)not null,  
  regione varchar(30)not null,  
  via varchar(30)not null,  
  nCivico varchar(5)not null )
```

```
CREATE TABLE richieste(  
  idRichiesta int not null PRIMARY KEY auto_increment,  
  rifUtente int not null,  
  oraInizioConsegna time not null,  
  oraFineConsegna time not null,  
  durataRichiesta time not null )
```

```
CREATE TABLE liste(  
  idLista int not null PRIMARY KEY auto_increment,  
  rifRichiesta int not null,  
  rifProdotto int not null,  
  Q.Tà int(10)not null )
```

```
ALTER TABLE richieste
```

ADD CONSTRAINT associazione1

FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifProdotto) REFERENCES prodotti(IdProdotto)

1)

GET /spesa/risposte				
Summary				
Elenco delle richieste completate				
Description				
Permette di avere l'elenco di tutte le richieste soddisfatte/completate da parte di un utente oppure solo una di queste				
▪ Produces: text/xml				
Parameters				
Name	Located In	Description	Required	Schema
rifUtente	Query	ID dell'utente del quale si vuole ottenere la lista di richieste completate	Yes	Xs:string
rifRichiesta	Query	ID della richiesta relativa ad una lista portata a termine/svolta da un utente	No	Xs:string
Lista	Query	Valore utilizzato per richiedere o meno anche la lista dei prodotti associata a tale richiesta	No	Xs:boolean
Responses				
Code	Description	Schema		
200	Elenco o una delle richieste completate da parte di un	<xs:complexType name="elencoRisposte"> <xs:sequence> <xs:element name="risposta" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence>		



		</xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>
<b>403</b>	Non autorizzato	
<b>406</b>	Parametri di input non validi	
<b>415</b>	valore del content-type non è quello atteso	
<b>500</b>	Internal Server Error	
<b>Default</b>	Errore non previsto	

### Esempi output proposto all'utente dal consumer:

Username	Nome	Cognome	CodiceFiscale	Regione	Via	nCivico
fraGali	Francesco	Galimberti	GLMFNC01A02B729Q	Lombardia	Giacomo Leopardi	5

oraInizio	oraFine	Durata	Lista					
20:41:18	21:41:18	01:00:00	Genere	Etichetta	Costo	Nome	Marca	Descrizione
			Alimentare	AAAAA	20.00	Pizza regina	Cameo	Pizza surgelata

### Esempi query SQL:

```

SELECT
Username,utenti.Nome,Cognome,CodiceFiscale,Regione,Via,nCivico,oraInizioConsegna,oraFineConsegna,
durataRichiesta,genere,etichetta, costo,prodotti.nome,marca,descrizione
FROM utenti,risposte,richieste,liste,prodotti
WHERE risposte.rifUtente=1 AND risposte.rifRichiesta=richieste.idRichiesta AND
richieste.rifUtente=utenti.idUtente AND richieste.idRichiesta=liste.rifRichiesta
AND liste.rifProdotto=prodotti.idProdotto

```

### Esempi create table SQL:

```

CREATE TABLE risposte (
  idRisposta int not null PRIMARY KEY auto_increment,
  rifUtente int not null,
  rifRichiesta int not null )

```

ALTER TABLE risposte  
 ADD CONSTRAINT associazione2  
 FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)

ALTER TABLE risposte  
 ADD CONSTRAINT associazione3  
 FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)

2)

GET /spesa/prodotto

### Summary

Elenco dei prodotti

### Description

Permette di ottenere un elenco di prodotti in base a determinati parametri

- Produces: text/xml

### Parameters

Name	Located In	Description	Required	Schema
Genere	Query	Genere del prodotto	No	Xs:string
Etichetta	Query	Etichetta del prodotto	No	Xs:string
Costo	Query	Costo del prodotto	No	Xs:double
Nome	Query	Nome del prodotto	No	Xs:string
Marca	Query	Marca del prodotto	No	Xs:string
Descrizione	Query	Descrizione del prodotto	No	Xs:string

### Responses

Code	Description	Schema
200	Elenco dei prodotti	<pre>&lt;xs:complexType name="return"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="prodotto" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="idProdotto" type="xs:int"/&gt;           &lt;xs:element name="Genere" type="xs:string"/&gt;           &lt;xs:element name="Etichetta" type="xs:string"/&gt;           &lt;xs:element name="Costo" type="xs:double"/&gt;           &lt;xs:element name="Nome" type="xs:string"/&gt;           &lt;xs:element name="Marca" type="xs:string"/&gt;           &lt;xs:element name="Descrizione" type="xs:string"/&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;</pre>

		</xs:sequence> </xs:complexType>															
403	Non autorizzato																
406	Parametri di input non validi																
415	valore del content-type non è quello atteso																
500	Internal Server Error																
Default	Errore non previsto																
<b>Esempi output proposto all’utente dal consumer:</b>																	
<table><tr><td>Genere</td><td>Etichetta</td><td>Costo</td><td>Nome</td><td>Marca</td><td>Descrizione</td></tr><tr><td>Alimentare</td><td>Aaaaa</td><td>2,50</td><td>Pizza margherita</td><td>Bbbb</td><td>Una pizza margherita senza glutine</td></tr></table>						Genere	Etichetta	Costo	Nome	Marca	Descrizione	Alimentare	Aaaaa	2,50	Pizza margherita	Bbbb	Una pizza margherita senza glutine
Genere	Etichetta	Costo	Nome	Marca	Descrizione												
Alimentare	Aaaaa	2,50	Pizza margherita	Bbbb	Una pizza margherita senza glutine												
<b>Esempi query SQL:</b>																	
SELECT idProdotto,Genere,Etichetta,Costo,Nome,Marca,Descrizione FROM prodotti																	
<b>Esempi create table SQL:</b>																	
CREATE TABLE prodotti( idProdotto int not null PRIMARY KEY auto_increment, genere varchar(30) not null, etichetta longtext not null, costo double(6,2)not null, nome varchar(50)not null, marca varchar(50)not null, descrizione longtext not null )																	

3)

POST /spesa/prodotto
<b>Summary</b>

Inserimento prodotto				
Description				
Permette di aggiungere un prodotto al database				
Consumes: text/xml				
Parameters				
Name	Located In	Description	Required	Schema
Prodotto	Body	Dati del prodotto da inserire	YES	<xs:complexType name="tipoProdotto"> <xs:sequence> <xs:element name="Genere" type="xs:string"/> <xs:element name="Etichetta" type="xs:string"/> <xs:element name="Costo" type="xs:double"/> <xs:element name="Nome" type="xs:string"/> <xs:element name="Marca" type="xs:string"/> <xs:element name="Descrizione" type="xs:string"/> </xs:sequence> </xs:complexType>  <xs:element name="prodotto" type="tipoProdotto"/>
Responses				
Code	Description	Schema		
200	Operazione eseguita correttamente			
403	Non autorizzato			
406	Parametri di input non validi			
415	valore del content-type non è quello atteso			
500	Internal Server Error			
Default	Errore non previsto			
Esempi output proposto all'utente dal consumer:				
➤ "Operazione andata a buon fine"				

➤ “Errore, operazione non completata”

### Esempi query SQL:

```
INSERT INTO prodotto (genere, etichetta, costo, nome, marca, descrizione) VALUES
($_POST[genereProdotto], $_POST[etichettaProdotto], $_POST[costoProdotto], $_POST[nomeProdotto],
$_POST[marcaProdotto], $_POST[descrizioneProdotto])
```

### Esempi create table SQL:

```
CREATE TABLE prodotti(
    idProdotto int not null PRIMARY KEY auto_increment,
    genere varchar(30) not null,
    etichetta longtext not null, costo double(6,2)not null,
    nome varchar(50)not null, marca varchar(50)not null,
    descrizione longtext not null )
)
```

4)

PUT /spesa/prodotto/{idProdotto}

### Summary

Modifica un prodotto

### Description

Permette la modifica di un prodotto specificando l'id del prodotto da modificare

- Consumes: text/xml

### Parameters

Name	Located In	Description	Required	Schema
idProdotto	Path	ID del prodotto	Yes	Xs:int
Prodotto	Body	Dati del prodotto modificati dall'utente	Yes	<pre>&lt;xs:complexType name="tipoProdotto"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Genere" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="Etichetta" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="Costo" type="xs:double" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="Nome" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>



				<pre>&lt;xs:element name="Marca" type="xs:string" minOccurs="0" maxOccurs="1"/&gt; &lt;xs:element name="Descrizione" type="xs:string" minOccurs="0" maxOccurs="1"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;  &lt;xs:element name="prodotto" type="tipoProdotto"/&gt;</pre>
<b>Responses</b>				
<b>Code</b>	<b>Description</b>	<b>Schema</b>		
<b>200</b>	Operazione eseguita correttamente			
<b>403</b>	Non autorizzato			
<b>406</b>	Parametri di input non validi			
<b>415</b>	valore del content-type non è quello atteso			
<b>500</b>	Internal Server Error			
<b>Default</b>	Errore non previsto			
Esempi output proposto all’utente dal consumer:				
<div>➤ “Operazione andata a buon fine”</div> <div>➤ “Errore, operazione non completata”</div>				
<b>Esempi query SQL:</b>				
UPDATE prodotto SET genere=\$_POST[genereProdotto], etichetta=\$_POST[etichettaProdotto], costo=\$_POST[costoProdotto], nome=\$_POST[nomeProdotto], marca=\$_POST[marcaProdotto], descrizione=\$_POST[descrizioneProdotto] WHERE idProdotto=\$_POST[idProdotto]				
<b>Esempi create table SQL:</b>				
CREATE TABLE prodotti( idProdotto int not null PRIMARY KEY auto_increment, genere varchar(30) not null, etichetta longtext not null, costo double(6,2)not null, nome varchar(50)not null, marca varchar(50)not null, descrizione longtext not null )				

)

5)

**DELETE** /spesa/prodotto/{idProdotto}

### Summary

Eliminazione di un prodotto

### Description

Permette di eliminare un dato prodotto

- Consumes: text/xml

### Parameters

Name	Located In	Description	Required	Schema
idProdotto	Path	ID del prodotto che si vuole eliminare	Yes	Xs:int

### Responses

Code	Description	Schema
<b>200</b>	Operazione eseguita correttamente	
<b>403</b>	Non autorizzato	
<b>406</b>	Parametri di input non validi	
<b>415</b>	valore del content-type non è quello atteso	
<b>500</b>	Internal Server Error	
<b>Default</b>	Errore non previsto	

### Esempi output proposto all'utente dal consumer:

- "Operazione andata a buon fine"
- "Errore, operazione non completata"

**Esempi query SQL:**

```
DELETE FROM prodotto WHERE idProdotto=[idProdotto]
```

**Esempi create table SQL:**

```
CREATE TABLE prodotti(  
    idProdotto int not null PRIMARY KEY auto_increment,  
    genere varchar(30) not null,  
    etichetta longtext not null,  
    costo double(6,2)not null,  
    nome varchar(50)not null,  
    marca varchar(50)not null,  
    descrizione longtext not null )  
)
```